Computers & Industrial Engineering 98 (2016) 373–383

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A Hungarian penalty-based construction algorithm to minimize makespan and total flow time in no-wait flow shops

Dipak Laha [a], Jatinder N.D. Gupta [b,*]

[a] Department of Mechanical Engineering, Jadavpur University, Kolkata, India
[b] College of Business Administration, University of Alabama in Huntsville, USA

ARTICLE INFO

Article history:
Received 5 December 2015
Received in revised form 2 June 2016
Accepted 3 June 2016
Available online 5 June 2016

Keywords:
Scheduling
No-wait flow shops
Makespan
Total flow time
Construction algorithm
Hungarian penalty based heuristic
Insertion heuristic
Traveling salesman problem
Optimization

ABSTRACT

This paper presents a penalty-based construction algorithm for the no-wait flow shop scheduling problem with the objective of minimizing makespan and total flow time of jobs. The proposed method, derived from Hungarian penalty method originally used for the classic assignment problem is employed to generate an initial schedule of jobs, which is further improved by an insertion technique to obtain an optimal or near-optimal schedule. The results of computational experiments on a large number of test problems show that the proposed method performs significantly better than the state-of-the-art procedures while requiring comparable computational effort.

1. Introduction

No-wait flow shop scheduling has long drawn the attention of researchers during last six decades due to its NP-hard class and practical importance. This type of problem occurs due to the processing characteristics of a product where it is needed continuous processing of jobs without interruption is needed, resulting in no-wait or continuous flow shop scheduling problems (Gupta, 1976; Reddi & Ramamoorthy, 1972; Wismer, 1972). It finds a wide range of applications in various industries and real-life applications (Framinan & Nagano, 2008; Hall & Sriskandarajah, 1996; Nagano, Silva, & Lorena, 2012).

Since no-wait flow shop scheduling is known to be NP-hard (Papadimitriou & Kanellakis, 1980; Röck, 1984), approximate optimization algorithms such as heuristics and metaheuristics are generally preferred to find good quality solutions (optimal or close to the optimal solutions) in relatively short computational time, especially for problems involving a large number of jobs. Heuristics are simple and usually require much less computational effort when compared with metaheuristics. On the other hand, metaheuristics, stochastic in nature are robust adaptive search optimization methods used for solving more difficult and complex problems.

A number of heuristics and metaheuristics in no-wait flow shop scheduling have been suggested in the literature mainly based on two optimization criterion: total flow time and makespan. Noteworthy heuristics with respect to total flow time objective have been proposed by Rajendran and Chaudhuri (1990), Bertolissi (2000), Aldowaisan and Allahverdi (2004), Framinan, Nagano, and Moccellin (2010), Laha, Gupta, and Sapkal (2014), and Laha and Sapkal (2014). Heuristics based on minimization of makespan have been studied by Bonney and Gundry (1976), King and Spachis (1980), Gangadharan and Rajendran (1993), Rajendran (1994), and Laha and Chakraborty (2009).

Another class of effective construction approximate algorithms such as farthest insertion, nearest neighbor, cheapest insertion, originally used for solving traveling salesman problems (TSP) have been studied in no-wait flow shop scheduling problems (Fink & Voß, 2003; Framinan & Nagano, 2008) as well. Bagchi, Gupta, and Sriskandarajah (2006) provided a comprehensive review of TSP based approaches along with the computational complexity for a variant of flow shop problems including the no-wait flow shops. Among these three construction algorithms, it has been shown that the farthest insertion method performs best based on a number of extensive and independent studies (Adrabinski & Syslo, 1983;

* Corresponding author.
E-mail addresses: dipaklaha_jume@yahoo.com (D. Laha), guptaj@uah.edu (J.N.D. Gupta).

http://dx.doi.org/10.1016/j.cie.2016.06.003
0360-8352/

Golden, Bodin, Doyle, & Stewart, 1980). However, for no-wait flow shop scheduling with the objective of minimizing total flow time, Fink and Voß (2003) have shown that cheapest insertion (they designated as Chins) algorithm performs much better than nearest neighbor (they designated as NN) algorithm. Apart from Chins and NN, they have also proposed some additional construction methods, combing a pilot method with NN and Chins such as Pilot-1-NN, Pilot-Chins, Pilot-1-Chins, Pilot-10-chins and Pilot-20-Chins, however at the cost of excessive computational times. Framinan and Nagano (2008) proposed their heuristic by applying the farthest insertion technique in no-wait flow shops to minimize makespan for generating initial schedule, followed by an improvement method of Framinan and Leisten (2003), which is originally used for the minimization of total flow time in permutation flow shop scheduling problems. Therefore, in this study, construction heuristics of Gangadharan and Rajendran (1993), heuristic of Rajendran (1994), heuristic of Framinan and Nagano (2008), nearest neighbor (NN), cheapest insertion (CI), and farthest insertion (FI) algorithms are considered for comparative analysis. It may be noted that, in general, NN, CI, and FI algorithms produce different solutions with different starting jobs. Therefore, when a further improved quality solution is required, it is better to run each of these algorithms starting once from each job of n jobs and obtain the best one among these n solutions, however, at the cost of some additional time complexity.

Metaheuristics have been applied to solve flow shop scheduling problems, namely the early works of Osman and Potts (1989) and Taillard (1990). For the no-wait flow shop scheduling problems, several metaheuristics have been studied, namely, genetic algorithm (Aldowaisan & Allahverdi, 2003; Chen, Neppalli, & Aljaber, 1996), simulated annealing (Fink & Voß, 2003), tabu search (Grabowski & Pempera, 2005), particle swarm optimization (Liu, Wang, & Jin, 2007; Pan, Tasgetiren, & Liang, 2008; Pan, Wang, Tasgetiren, & Zhao, 2008), artificial immune system (Kumar, Prakash, Shankar, & Tiwari, 2006), hybrid tabu search and particle swarm optimization (Samarghandi & ElMekkawy, 2012), ant colony optimization algorithm (Shyu, Lin, & Yin, 2004), hybrid genetic algorithm and variable neighborhood search algorithm (Jarboui, Eddaly, & Siarry, 2011), and differential evolution (Qian, Wang, Hub, Huang, & Wang, 2009). Constructive heuristics are frequently used for generating good starting solutions for the metaheuristics.

In this paper, we propose a penalty-based algorithm for the no-wait flow shop-scheduling problem to minimize makespan with the goal of finding better makespan or total flow time schedules. The proposed method uses Hungarian method-based technique to obtain an initial schedule of jobs and is then improved by employing an insertion method. The proposed method is compared with the best-known algorithms to demonstrate its effectiveness in finding a better quality solution in comparable CPU time.

The remainder of this paper is structured as: Section 2 gives the problem description. A brief contribution of the existing heuristics are described in Section 3. The proposed algorithm is presented in Section 4. A numerical example to illustrate the proposed method is given in Section 5. Section 6 provides the results of computational experimentation and conclusions are made in Section 7.

2. Problem description

Given the processing time p_{ij} of job i on machine j in the no-wait flow shop scheduling, each of n jobs is processed on m machines in the same technological order without preemption and interruption on or between any two consecutive machines. The problem is to determine a schedule of n jobs that minimizes the makespan and the total flow time. Let $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ represent the schedule of n jobs to be processed on m machines, and $d(i,k)$ the minimum delay on the first machine between the start of job i and the start of job k (required because of the no-wait restriction). Also, let $p(\sigma_i, j)$ represent the processing time on machine j of the job in the ith position of a given schedule, and let $d(\sigma_{i-1}, \sigma_i)$ denote the minimum delay on the first machine between the start of two consecutive jobs found in the $(i-1)$th and ith position of the schedule.

The m-machine no-wait flow shop scheduling problem to minimize makespan is denoted by $F_m|no\text{-}wait|C_{max}$. Gupta (1976) showed that only permutation schedules are feasible. Therefore, the aim to search for the optimal or near-optimal schedule for the $F_m|no\text{-}wait|C_{max}$ problem is limited to permutation schedules only. Further, based on the developments in Gupta (1976), $d(i,k)$ can be determined from the following expression:

$$d(i,k) = p_{i1} + \max\left(\max_{2\leqslant h\leqslant m}\left[\sum_{s=2}^{h} p_{is} - \sum_{s=1}^{h-1} p_{ks}\right]; 0\right) \quad (1)$$

where $d_{0j} = 0$ for all $j \in N$. The delay matrix $D = [d(i,k)]$ gives all the d_{ij} values between the start of any two consecutive jobs i and j $(i \neq j)$ in a given sequence of n jobs to determine the objective function MS.

Let Π denote the set of all $n!$ possible permutation schedules in the search space for the no-wait flow shop problem. Let σ represent a permutation schedule, $\sigma \epsilon \Pi$ and $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. Then, the completion time of σ_j on machine m, $C(\sigma_j, m)$ is obtained as:

$$C(\sigma_j, m) = \sum_{k=1}^{j} d(\sigma_{k-1}, \sigma_k) + \sum_{s=1}^{m} p(\sigma_j, s) \quad (2)$$

Therefore, using Eq. (2), the makespan of the schedule σ, $MS(\sigma)$ *in* the no-wait flow shop scheduling is given by

$$MS(\sigma) = \sum_{i=2}^{n} d(\sigma_{i-1}, \sigma_i) + \sum_{j=1}^{m} p(\sigma_n, j) \quad (3)$$

Also, using Eq. (2), total flow time for schedule σ can be obtained as:

$$TFT(\sigma) = \sum_{j=1}^{n} C(\sigma_j, m) = \sum_{j=2}^{n} (n+1-j) d(\sigma_{j-1}, \sigma_j) + \sum_{j=1}^{n}\sum_{s=1}^{m} p_{js} \quad (4)$$

Then, the $F_m|no\text{-}wait|C_{max}$ problem is to find a permutation schedule $\sigma^* \epsilon \Pi$ such that:

$$MS(\sigma^*) = \min_{\sigma \epsilon \Pi} MS(\sigma) \quad (5)$$

Similarly, for the $F_m|no\text{-}wait|\sum C_j$ problem is to find a permutation schedule $\sigma^* \epsilon \Pi$ such that:

$$TFT(\sigma^*) = \min_{\sigma \epsilon \Pi} TFT(\sigma) \quad (6)$$

It may be noted that since the release times of the jobs are all zeros, the completion time criterion is equivalent to the MS criterion and the total completion time criterion is equivalent to the TFT criterion. The complexity of the delay matrix D in Eq. (1) is $O(n^2 m)$ and it is executed only once. Then the makespan as well as total flow time of any permutation schedule σ can be executed in $O(n)$ computational effort. Therefore, the complexity of the $n!$ permutation schedules is $O(n!n + n^2 m)$, which is exponential and hence are not feasible to solve problems involving jobs even 10 or more jobs.

3. Relevant heuristics

In this paper, we will consider the following seven best-known construction heuristics for comparative performance evaluation such as the heuristics of Gangadharan and Rajendran (1993), heuristic of Rajendran (1994), heuristic of Framinan and Nagano (2008), heuristic of Laha et al. (2014), nearest neighbor, farthest